

REVISITING THE TORT SOLUTIONS TO THE NEA SUITE OF BENCHMARKS FOR 3D TRANSPORT METHODS AND CODES OVER A RANGE IN PARAMETER SPACE

Kursat B. Bekar*

Radiation Transport and Criticality Group
Oak Ridge National Laboratory, Oak Ridge, TN 37831-6170
bekarkb@ornl.gov

Yousry Y. Azmy

Department of Nuclear Engineering
North Carolina State University, Raleigh, NC 27695
yyazmy@ncsu.edu

ABSTRACT

Improved TORT solutions to the 3D transport codes' suite of benchmarks exercise are presented in this study. Preliminary TORT solutions to this benchmark indicate that the majority of benchmark quantities for most benchmark cases are computed with good accuracy, and that accuracy improves with model refinement. However, TORT fails to compute accurate results for some benchmark cases with aspect ratios drastically different from 1, possibly due to ray effects. In this work, we employ the standard approach of splitting the solution to the transport equation into an uncollided flux and a fully collided flux via the code sequence GRTUNCL3D and TORT to mitigate ray effects. The results of this code sequence presented in this paper show that the accuracy of most benchmark cases improved substantially. Furthermore, the iterative convergence problems reported for the preliminary TORT solutions have been resolved by bringing the computational cells' aspect ratio closer to unity and, more importantly, by using 64-bit arithmetic precision in the calculation sequence. Results of this study are also reported.

Key Words: TORT, transport codes' benchmark, GRTUNCL3D

1. INTRODUCTION

The 3-D transport codes' benchmark exercise, *Benchmarking the Accuracy of Solution of 3-dimensional Transport Codes and Methods over a Range in Parameter Space*, was adopted by NEA/OECD to test the accuracy and performance of three-dimensional transport codes and methods over a wide range in parameter space spanned by the problem geometry and nuclear data [1].

At the earlier blind stage of the benchmark exercise, preliminary TORT [2] solutions to this benchmark, and the effect on solution accuracy of various computational model refinements, namely, order and type of the angular quadratures, and the computational mesh size were reported [3,4]. For some benchmark cases, the specified 10^{-4} convergence criterion was not met in 100 iterations, but a respectable 2×10^{-3} criterion was achieved. Possible ray effects in the discrete-ordinates solutions is at least partially to blame for this convergence failure. Preliminary

*Corresponding author

results, reported in Ref. 3, indicate that while the solutions for most cases in the suite of benchmarks as computed by TORT are reasonably accurate, and possess decreasing error with model refinement, some of the benchmark cases with aspect ratios drastically different from 1 failed to produce accurate results, possibly due to ray effects.

In this work we explore methods to reduce adverse ray effects by employing the standard approach of splitting the solution to the transport equation into an uncollided flux and a fully collided flux. The former is computed via the ray-tracing approach implemented in the code GRTUNCL3D [5], and the latter is computed with TORT driven by a first collision source. Tests of this approach illustrate the improved accuracy of several benchmark quantities that suffered large errors in the straight discrete-ordinates calculation.

Although the use of a GRTUNCL3D-TORT sequence improves the solution accuracy by reducing ray effects for most cases, it is insufficient by itself to resolve the convergence problem entirely. For this reason, different strategies were also followed to achieve the specified convergence criterion for all benchmark cases. The results of these approaches, including remeshing to bring the cell aspect ratio closer to unity and employing 64-bit arithmetic precision, are also reported in this paper.

2. TORT MODELS FOR BENCHMARK PROBLEM

For the benchmark calculations, specifications of four TORT models were reported in detail in Ref. 3.

In the transport model, a uniform mesh structure was used along each coordinate axis. For the coarsest mesh of the TORT model, mesh numbers were set at $40 \times 40 \times 40$ for each case in the suite of benchmarks by using the parameters L and γ , and the boundaries of the sub-regions over which the scalar flux will be computed. Models with finer mesh structures were obtained by dividing each cell's size in each dimension by 2, 3, and 4, thus producing $80 \times 80 \times 80$, $120 \times 120 \times 120$, and $160 \times 160 \times 160$ uniform meshes, respectively.

In the TORT calculations, Square Legendre-Chebyshev (SLC) angular quadratures were used. In addition the spatial and angular discretizations were refined simultaneously; hence, the number of non-zero-weight angles in SLC was selected as 200, 392, 648, and 800 from the coarsest model to the finest model, respectively.

The θ -weighted method was used with $\theta = 0.9$ (the default) in the TORT calculations, and 10^{-4} convergence of the scalar flux was required. The limit on the number of inner iterations was set to 100.

3. NUMERICAL RESULTS

The four models described above, each representing a different level of model refinement, were solved with TORT for all 729 benchmark cases via the benchrun-1.2 script [6]. Then, all benchmark quantities computed on the four levels of model refinement are compared to the MCNP5 [7] reference solution. For some TORT cases, especially in the finest model, the strict

10^{-4} convergence criterion was not met in 100 iterations, but a respectable 2×10^{-3} criterion was achieved.

Samples of the preliminary results are shown in Fig. 1 where a comparison between the relative errors for each of the 729 benchmark cases computed on the four refinement levels of the TORT model is illustrated for four representative benchmark quantities; see Ref. 1 for specifications of these quantities. Note the fixed scale along the vertical axis intended to underscore the non-uniform dependence of the error magnitude on the benchmark quantities considered. In Figs. 1.a and 1.b, TORT produced highly accurate results. Although for most benchmark cases the relative error decreases when the computational model is refined, for some benchmark quantities, 2.e and 3.m, shown in Figs. 1.c and 1.d, respectively, the relative error is larger than 20% for some cases in the benchmark suite, and in some instances the error decrease is not monotonic with model refinement.

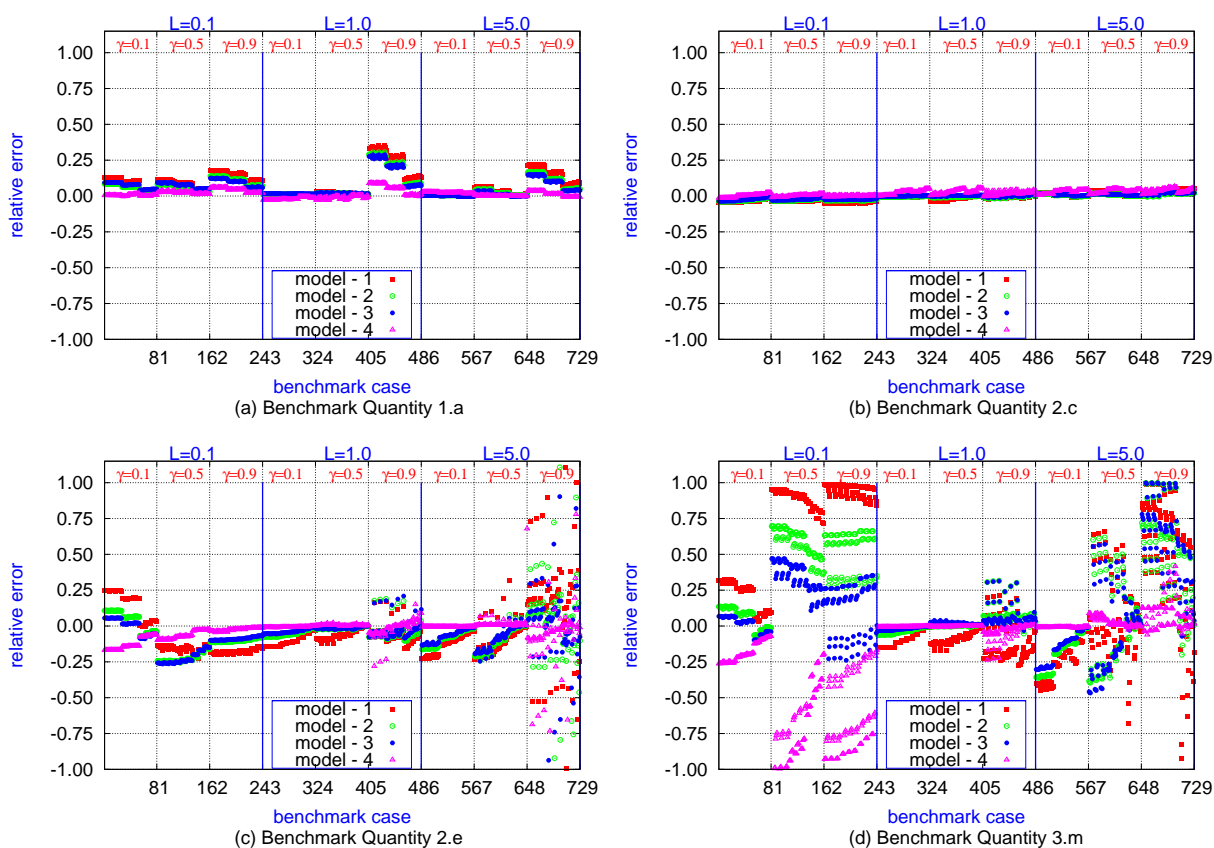


Figure 1. Relative errors calculated for some benchmark quantities over all cases in the benchmark suite on the four TORT models (model-1 the coarsest, model-4 the finest).

The region in parameter space where these difficulties are most noticeable is characterized by aspect ratios far from 1, wherein surface areas or region volumes used for averaging the scalar flux or net current are too small and too far from the small source region. Basically this results in severe ray effects that we propose will be eliminated by computing the uncollided flux and the first collision source by GRTUNCL3D, then the latter is used in a subsequent TORT run to

compute the fully collided flux. After obtaining a converged solution from TORT, the resulting collided flux and GRTUNCL3D's uncollided flux are added to generate the total fluxes. GRTUNCL3D uses a semi-analytic method to generate uncollided flux distributions and spatial, energy(in case of multigroup applications), and angular-dependent first collision sources using a set of point source declarations in its input file that represent the source distribution in the original problem configuration, and geometry information provided in the TORT input file. In other words, GRTUNCL3D executes off of the TORT input file plus additional source information in a separate input file.

The current version of GRTUNCL3D does not compute the uncollided fluxes at the cell boundaries; thus, leakage across internal and external boundaries, as many of the benchmark quantities represent, cannot be calculated by the method outlined above. Therefore, GRTUNCL3D was modified to enable this feature to provide uncollided angular fluxes not only at the cell centers but also at the cell boundaries, and these can be post-processed into uncollided net leakages for the purposes of this benchmark exercise.

The results of the GRTUNCL3D-TORT sequence, which is denoted as *GTort* in this paper, shown in Fig.2, were obtained by using the coarsest TORT model ($40 \times 40 \times 40$, with SLC-10, 200 non-zero-weight angles) for a formerly failed case (case-331111, see Ref. 6 for the naming convention for the benchmark cases) to improve the accuracy of the TORT results. Figure 2.b confirms the existence of strong ray effects in the preliminary TORT solutions. Generating a first collision source and repeating the TORT calculations for the same case eliminates the primary ray effects as illustrated in Fig. 2.c and exhibits better agreement with the MCNP distribution shown in Fig. 2.a.

Table I presents both the net leakages computed across the surfaces of inner and outer regions, and the scalar fluxes averaged over the defined regions in the benchmark that were computed by straight TORT, and also by the *GTort* sequence; then both solutions are compared to the reference

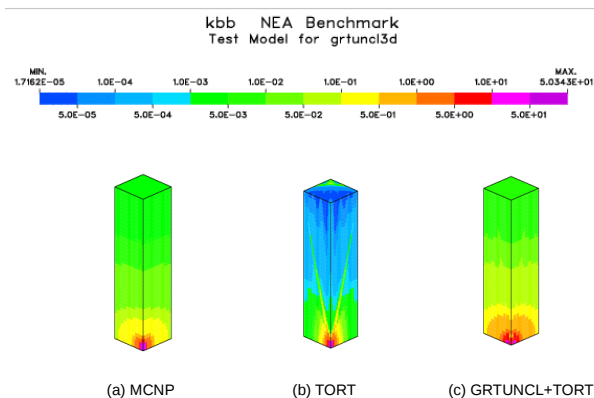


Figure 2. Scalar flux distribution over the problem domain for the case with $L = 5.0$, $\gamma = 0.9$, $\sigma_1 = \sigma_2 = 0.1$, and $c_1 = c_2 = 0.5$ (see Ref. 1 for these definitions).

Table I. Results of some quantities of TORT model-1 ($40 \times 40 \times 40$, $SLC - 10$) for the benchmark case $L = 5.0$, $\gamma = 0.9$, $\sigma_1 = \sigma_2 = 0.1$, and $c_1 = c_2 = 0.5$

benchmark quantity	ADVANTG/ MCNP5	TORT		GTtort		MGTort	
		value	% diff.	value	% diff.	value	% diff.
1.a	1.24E-01	1.09E-01	-12.37	1.94E-01	56.28	1.09E-01	-12.37
1.b	3.31E-02	3.60E-02	8.77	3.34E-02	1.11	3.60E-02	8.77
2.a	-3.44E-01	-3.50E-01	1.70	-2.32E-01	-32.48	-3.46E-01	0.68
2.b	6.28E-02	5.88E-02	-6.36	6.36E-02	1.27	6.36E-02	1.27
2.c	-1.70E-01	-1.67E-01	-1.60	-1.51E-01	-10.94	-1.70E-01	0.02
2.d	2.03E-03	1.56E-03	-23.37	2.04E-03	0.02	2.04E-03	0.02
2.e	1.52E-02	1.55E-02	1.95	1.31E-02	-13.60	1.31E-02	-13.60
2.f	4.60E-02	4.96E-02	7.92	4.67E-02	1.47	4.67E-02	1.47
2.g	6.95E-02	7.64E-02	10.04	7.49E-02	7.91	7.49E-02	7.91
2.h	1.87E-03	1.61E-03	-13.89	1.87E-03	0.17	1.87E-03	0.17
3.a	5.08E+01	5.03E+01	-0.81	6.81E-01	-98.66	5.03E+01	-0.81
3.b	1.10E-01	1.21E-01	9.65	1.11E-01	0.68	1.11E-01	0.68
3.c	4.93E-03	5.92E-03	20.01	5.03E-03	1.98	5.03E-03	1.98
3.d	2.13E-03	3.26E-03	52.83	2.26E-03	6.20	2.26E-03	6.20
3.e	7.89E-02	1.17E-03	-98.51	7.89E-02	0.03	7.89E-02	0.03
3.f	3.81E-02	2.40E-01	528.67	4.11E-02	7.83	4.11E-02	7.83
3.g	2.19E-03	1.07E-04	-95.09	2.18E-03	-0.32	2.18E-03	-0.32
3.h	2.08E-03	6.10E-05	-97.07	2.09E-03	0.69	2.09E-03	0.69
3.i	1.98E-03	2.91E-03	47.27	2.11E-03	6.73	2.11E-03	6.73
3.j	4.95E-02	5.10E-02	3.11	5.02E-02	1.43	5.02E-02	1.43
3.k	3.46E-02	3.75E-02	8.29	3.51E-02	1.45	3.51E-02	1.45
3.l	5.43E-03	1.50E-03	-72.36	5.40E-03	-0.63	5.40E-03	-0.63
3.m	5.17E-03	1.00E-02	93.23	5.32E-03	2.79	5.32E-03	2.79

solutions for such benchmark quantities. The preliminary reference solution set was obtained by MCNP5, but this set exhibited many failures such as either 0-tally scoring or poor statistics in the results of the some benchmark quantities. Subsequently, the reference solutions were improved via Oak Ridge National Laboratory's ADVANTG code sequence [8]. ADVANTG uses TORT driven cell-averaged scalar flux distributions to generate Monte Carlo weight windows parameters by implementing FW-CADIS methodology [9]. Then, MCNP5 computes the benchmark quantities for all benchmark cases using the generated weight windows parameters.

For the selected case in Table I, the difference between TORT's values and the ADVANTG/MCNP5 reference results are extremely large for all benchmark quantities except the quantities computed near the source region where GRTUNCL3D is known to grossly underestimate the uncollided flux. Hence, in order to improve solutions everywhere, we modified the GTort sequence such that (1) straight TORT computes the cell-averaged fluxes and boundary fluxes, (2) subsequently a GTort sequence computes same, and (3) a post-processor reads fluxes from the straight TORT's data file(s) for only the source region (cells in the source region, and the surfaces of cell which sit on the source boundary), and reads fluxes from GTort sequence's data file(s) for the whole geometry except the source region. The results of ADVANTG/MCNP5, TORT, GTort sequence and the modified GTort sequence denoted MGTort are presented in Table I for this benchmark case (case number=331111). The results in Table I indicate that the GTort computational sequence improves the accuracy for most quantities for this benchmark case, thus resolving previously observed difficulties. However, in the source region GRTUNCL3D produces unacceptable results due to its inability to accurately perform ray-tracing within a source cell. By introducing the MGTort sequence, this problem was fixed and reasonable results were obtained. For the case shown in Table I the MGTort sequence results in the reduction of solution error for the coarsest model from the 100% range to below 8% for most benchmark quantities except the benchmark quantity 2.e. We are currently investigating the reason for the discrepancy in this benchmark quantity.

The same calculation sequences were performed with the other three TORT computational models for the benchmark cases with $\gamma = 0.9$ that correspond to a relatively smaller source region, and therefore ray effects might be more pronounced for these cases. In Figs. 3 and 4, comparison between the relative errors for the benchmark cases with $\gamma = 0.9$ computed by TORT and by the computational sequence MGTort on the four TORT computational models is illustrated for two representative benchmark quantities. The results show that the proposed computational sequence, MGTort, reduces the solution error significantly for the vast majority of benchmark cases. However, for the benchmark cases with $L = 0.1$, this statement is not completely correct for the model-3 because the results of model-3 are worse than the other two coarsest models, model-1 and model-2, which is unexpected. A case study is also being conducted to identify and fix the problem for this model for these benchmark cases.

In the initial stage of these calculations, although the MGTort sequence improved the solution accuracy for the benchmark cases of which solutions were inaccurate due to ray effects, it could not help to resolve the iterative convergence problem completely for this set. Moreover some benchmark cases computed by straight TORT that converged the inner iterations to the set tight criterion did not converge in the MGTort sequence. In order to determine the source of this problem, a comprehensive analysis was performed on these failed cases. One attempt was to modify the mesh so as to use cubic cells (unit aspect ratio) to resolve this problem since the

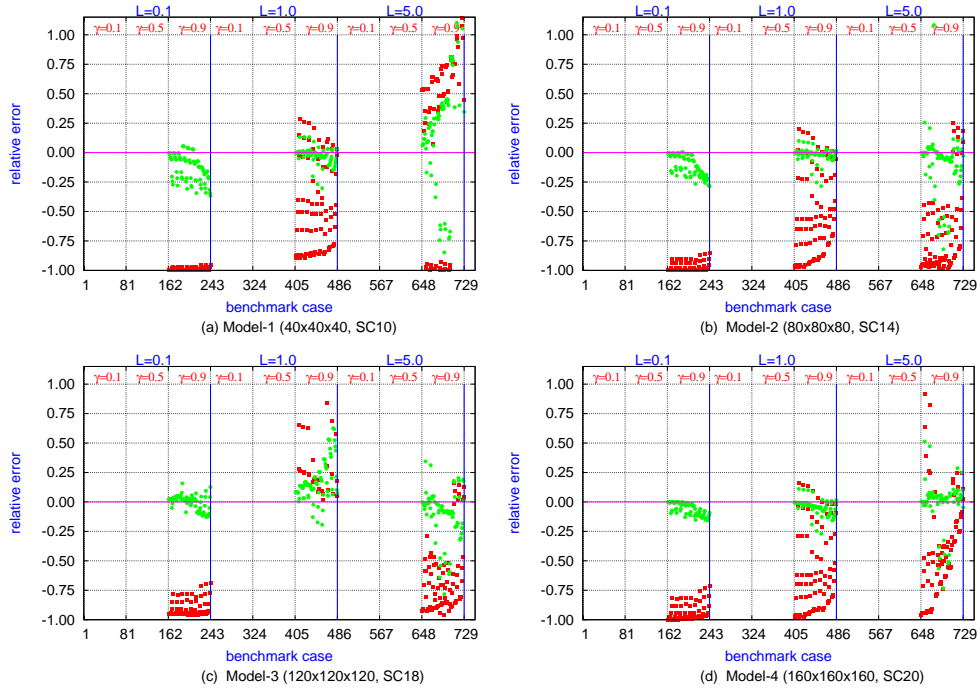


Figure 3. Relative errors calculated for the benchmark quantity 3.d over all cases ($\gamma = 0.9$) in the benchmark suite on the four computational models. In these figures red and green points represent TORT and GTort, respectively.

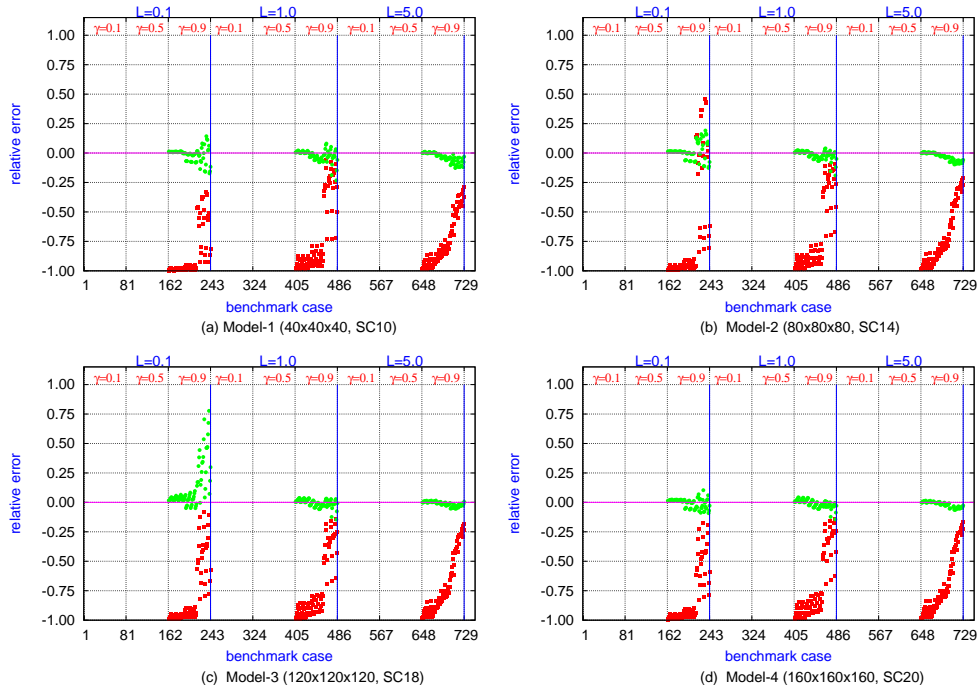


Figure 4. Relative errors calculated for the benchmark quantity 3.e over all cases ($\gamma = 0.9$) in the benchmark suite on the four computational models. In these figures red and green points represent TORT and GTort, respectively.

non-linear θ -weighted method works well if the cell aspect ratio is close to 1. The results of this analysis show that using $400 \times 400 \times 40$, $40 \times 40 \times 40$, and $40 \times 40 \times 200$ mesh structures for the benchmark cases $L = 0.1$, $L = 1.0$ and $L = 5.0$ solves the convergence problem for all the benchmark cases except five cases that were computed by TORT. For the calculations with the MGTort sequence, this approach did not help as it was expected. Because the uncollided component of flux determined by GRTUNCL3D is the dominant part of the flux, the collided flux comprises positive numbers that are too small due to this separation. Therefore, the convergence problem becomes evident for the TORT calculation in the MGTort sequence. In addition, the use of cubic cells is not always a realistic option since TORT and GRTUNCL3D with the finest computational model need an extremely large amount of memory and disk space to perform their calculations. In another attempt to solve this problem, both TORT and GRTUNCL3D were recompiled, making the default integer and real variables double-precision (8byte long), and the resulting executables were re-tested with the MGTort sequence.

Although the main disadvantages of double-precision codes are the longer computation time and the larger memory and disk space requirements, the results indicate that all cases for the four TORT computational models converged the inner iterations to the tight criterion of 10^{-4} . A sample comparison for the change in convergence rate for the MGTort sequences with both single and double-precision codes is depicted in Fig. 5 for the benchmark case 323133.

Double-precision (DP) TORT converges in 48 inner iterations for this sample benchmark case, whereas single-precision (SP) TORT initially approaches convergence to an iterative residual value almost equal to 2×10^{-3} , and then oscillates in an interval between -2×10^{-3} and 2×10^{-3} .

In order to make a performance comparison, the execution times of the two TORT executables with model-2 for all benchmark cases are presented in Fig. 6. The results show that the

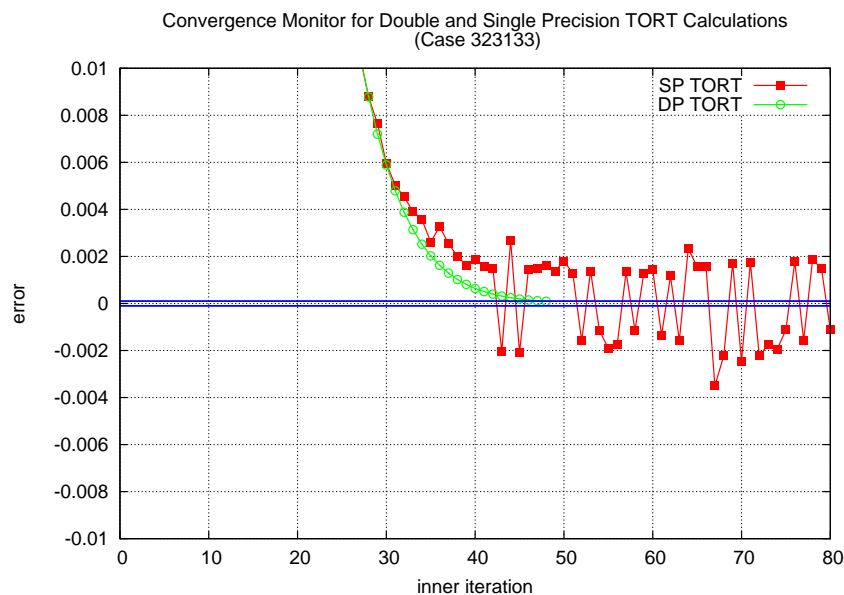


Figure 5. Change in convergence rate for two TORT executables for the benchmark case 323133, $L = 5.0$, $\gamma = 0.5$, $\sigma_1 = \sigma_2 = 5.0$, $c_1 = 0.5$ and $c_2 = 1.0$.

computation time for both executables are almost comparable for model-2 except for some benchmark cases. For these cases, the single-precision TORT execution took longer than the double-precision TORT since the single-precision TORT did not converge in 100 inner iterations, whereas the double-precision TORT converged rapidly. It is a fact that, for finer models, the double-precision TORT slows down for most of the benchmark cases since the amount of I/O and physical memory overhead increases by a factor of two.

It is worth noting here that the results of the MGTort sequence presented in this paper were obtained by implementing double-precision codes into the sequence. In other words, all benchmark cases computed by the MGTort sequence converged to the 10^{-4} criterion in a reasonable number of inner iterations.

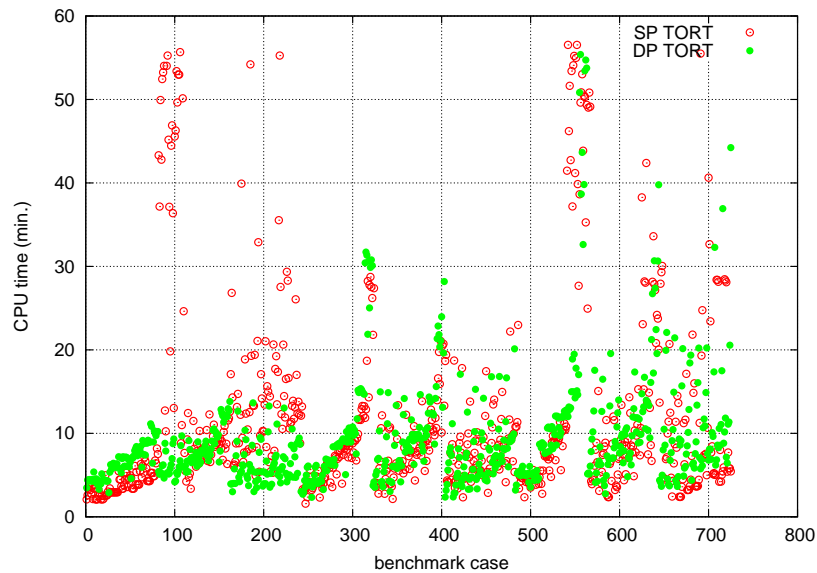


Figure 6. Comparison of execution times for two TORT executables for the benchmark case 323133, $L = 5.0$, $\gamma = 0.5$, $\sigma_1 = \sigma_2 = 5.0$, $c_1 = 0.5$ and $c_2 = 1.0$.

4. CONCLUSIONS

Based on the results of our preliminary numerical experiments, we conclude that the solutions for most cases in the suite of benchmarks as computed by TORT are reasonably accurate. For the benchmark cases and benchmark quantities where this is not true, we found that generating the first collision source and supplying this source to drive the TORT calculations of the fully collided flux yields more accurate results. In addition, the iterative convergence problems encountered in the preliminary numerical experiments and in the calculations with MGTort sequence were completely resolved by introducing double-precision versions of TORT and GRTUNCL3D and deploying them in the code sequence.

The improved solution set indicates that the TORT results (via MGTort sequence) and the ADVANTG/MCNP5 reference results are in good agreement for most of the benchmark quantities for most of the benchmark cases.

REFERENCES

- [1] Y. Y. Azmy, "Benchmarking the Accuracy of Solution of 3-dimensional Transport Codes and Methods over a Range in Parameter Space," NEA/NSC/DOC (2007)1/REV1 (2007).
- [2] W. A. Rhoades and D. B. Simpson, *The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code*, Computational Physics and Engineering Division, ORNL (1997).
- [3] K. B. Bekar, and Y. Y. Azmy, "TORT Solutions to the NEA Suite of Benchmarks for 3D Transport Methods and Codes over a Range in Parameter Space", *PHYSOR-2008, International Conference on Reactor Physics, Nuclear Power: A Sustainable Resource*, Interlaken, Switzerland (2008).
- [4] K. B. Bekar, and Y. Y. Azmy, "TORT Solutions to the NEA Suite of Benchmarks for 3D Transport Methods and Codes over a Range in Parameter Space", *Annals of Nuclear Energy*, **36**, pp 368-374 (2009).
- [5] J. O. Johnson (Ed.), *GRTUNCL3D, A Discontinuous Mesh Three-Dimensional First Collision Source Code*, ORNL (2004).
- [6] K. B. Bekar, "benchrun Manual", <http://www.nea.fr/html/science/eg3drtb/benchrun-1.2/Readme> (2007).
- [7] X-5 Monte Carlo Team, *MCNP-A General Monte Carlo N-Particle Transport Code, Version 5*, Manual, Los Alamos National Laboratory (2003).
- [8] J. C. Wagner, "An Automated Deterministic Variance Reduction Generator for Monte Carlo Shielding Applications," in *Proc. of ANS 12th Biennial Topical Meeting of the Radiation Protection and Shielding Division*, Santa Fe, New Mexico, April 14-18 (2002).
- [9] J. C. Wagner, E. D. Blakeman, and D. E. Peplow, "Forward-Weighted CADIS Method for Global Variance Reduction", *Trans. Am. Nucl. Soc.* **97**, pp. 630-633 (2007).